Viola systems

Embedding The Internet

Viola ESERV-10 S

for SATELLINE radio modems

User's Manual

Version 2.0, March 2002

1 Starter Kit contents and functional environment

1.1 Contents

Table 1. Starter Kit contents

Components		Number
Hardware	ESERV-10 S Internet Module	1
	RS232 Nullmodem cable	1
	Cross-connected Ethernet cable (RJ45)	1
Software	CD including ESERV-10 S control software, documentation and example programs.	1

1.2 Functional environment

- Operating voltage: 7 … 15 VDC or 5 VDC
- Current consumption: 180 mA (max), 100 mA (typ)
- Humidity: 0 … 95%
- Temperature: 0 … 60 °C

1.3 Technical support

Satel Oy
Meriniitynkatu 17
24100 Salo
Finland

Tel. +358 2 777 7800
Fax. +358 2 777 7810

Email: technicalsupport@satel.fi
http://www.satel.fi

2 Overview

Remark: ESERV-10 S is special module designed to be used with a SATELLINE radio modems.

ESERV-10 is a connection module utilizing Internet protocols (TCP/IP). This module device can be connected to the Internet or to a local network through a serial port without any need to modify existing hardware or software. With ESERV-10 it is possible for manufacturers and system suppliers to connect their products, for example home and industrial automation devices, measuring devices and sensors through Ethernet to Internet. Devices can be controlled and monitored using a standard Web browser (Internet Explorer, Netscape Navigator), a TELNET connection or some other programs with a TCP/IP connection.

For example ESERV-10 can be used in cases of:

- Remote reading of access control and work time monitoring systems
- Software updates and parametering over the Internet
- Motion sensors
- Remote control of display boards
- Global follow-up of systems
- Replacement of serial cable network with ready-made Ethernet network

ESERV-10 includes the core protocols of TCP/IP, and a built-in Web server as a graphic Java applet downloading from server to browser can function as the device's virtual user interface. ESERV-10 can handle all Internet related functions by acting as a modem towards the device connected to a serial port. In practice the serial device doesn't require any modification.

Table 2 shows the features of ESERV-10 in a nutshell:

Table 2. Features of ESERV-10

Feature	
Processor	16 bit RISC
Network connection	10Base-T Ethernet (IEEE 802.3)
Serial device connection	1 channel RS-232, 1 200 ... 38 400 bps
Memory	130 kB flash-memory for Web pages, updated over the Internet
Internet protocols supported	HTTP 1.1, BOOTP, TFTP
	TCP (6 connections), UDP
	IP, ICMP, ARP
	Ethernet (IEEE 802.3)
Control software for Windows 95/98/NT/2000	Device's control database
	BOOTP server for set up of IP addresses
	TFTP for transferring of Web pages

3 Introduction to the ESERV-10 module

The interfaces of ESERV-10 board can be seen in Figure 1. Figure 2 is the functional block diagram.

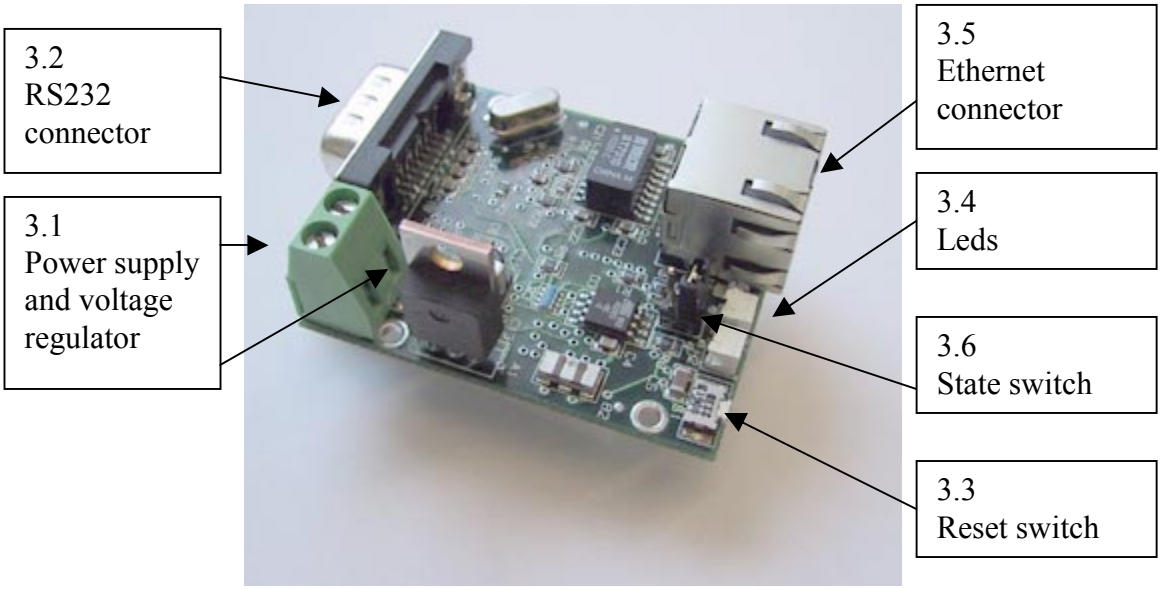


Figure 1. ESERV-10 interfaces

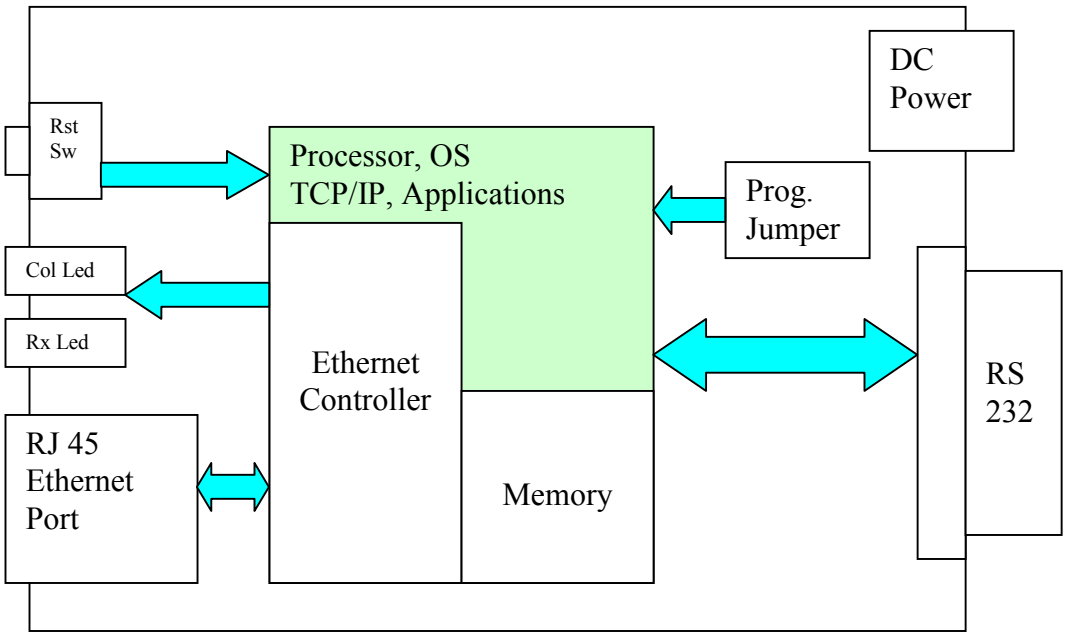


Figure 2. Functional block diagram

3.1 Power supply

Power supply connector is a screw connector for 1,5 mm^2 wires. The connector pin configuration is in Figure 3.

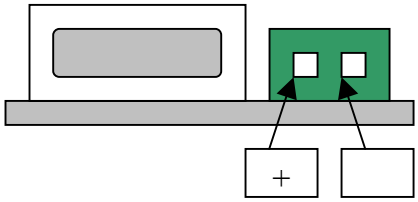


Figure 3. Power supply

- Operating voltage 7 to 15 VDC (optional 5 VDC).
- 180 mA max.
- Protected against improper coupling

3.2 Serial port (RS232 connector)

Serial connector is a 9-pin male connector (DB9). The pin configuration (Figure 4) is the same as for a PC (DTE). The Starter Kit includes a nullmodem cable, which is used to connect the ESERV-10 to the target device or a PC. The ESERV-10 supports CTS/RTS flow control.

As a manufacturing option it is possible to implement power supply by using the not connected pins (1 and 9) of the serial port.

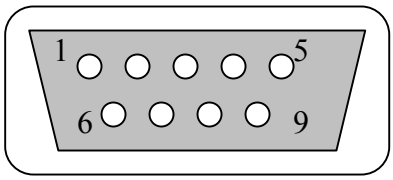


Figure 4. Pin configuration of RS232 connector

Table 3. Pin configuration of RS232 connector

Pin number	Term	Function	Note
1	NC	Not connected/ Power supply	Pin can be used as an optional power supply (+) for ESERV-10 or the target device
2	RXD	Receive data	Input
3	TXD	Transmit data	Output
4	NC	Not connected	
5	GND	Signal ground	
6	NC	Not connected	
7	RTS	Ready To Send	Handshake output
8	CTS	Clear To Send	Handshake input
9	NC	Not connected/ Power supply	Pin can be used as an optional power supply (GND) for ESERV-10 or the target device

3.3 Reset switch

The Ethernet controller and processor can be initialized by pressing the reset switch. It is used when updating the internal software of ESERV-10.

3.4 Leds

The two leds (shown in Figure 5) in the front panel of ESERV-10 board function as described in Table 4.

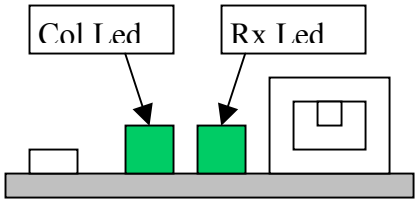


Figure 5. Leds

Table 4. Functioning of the leds

Led	Functioning
Collision Led (Col Led)	This led is normally off. It blinks when there is a collision in ESERV-10's Ethernet network segment. In case of a collision, ESERV-10 retransmits the packet max. 16 times. In a properly designed network these collisions take place very seldom. Frequently occurring collisions indicate that the network segment is heavily loaded.
Receiving Led (Rx Led)	This led is on when the operating voltage is connected. It blinks while receiving a packet from an Ethernet network.

3.5 Ethernet connector

In the front panel of ESERV-10 there is a RJ45 connector for Ethernet connection. ESERV-10 can be connected with a direct Ethernet cable (not included in the Starter Kit) to a local network's hub or switch. Maximum distance between the board and the hub/switch is 100m. ESERV-10 can be connected directly to a PC network interface card with the included cross-connected Ethernet cable for testing.

The included cross-connected cable with the red connectors is only suited for connecting the ESERV-10 board and a PC's network interface card. When connecting to a local network a direct Ethernet cable should be used.

3.6 State switch

State switch/jumper is used to set the ESERV-10 into the programming state when updating ESERV-10's internal software through a serial port. Normally the switch should stay open.

On the CD included in the Starter Kit there is a separate PC program and documentation concerning the software updating. Software updates are available from Viola's Web page http://www.violasystems.com.

3.7 Internal software

ESERV-10's internal software consists of Viola's IETF – (Internet Engineering Task Force – Community responsible for Internet's standardization) compatible TCP/IP stack, embedded HTTP/Web server and application software for serial device connectivity.

Table 5. Internal protocol implementation

OSI Layer	ESERV-10 implementation					
7 – Application	HTTP 1.1	Serial port	BOOTP	TFTP	Serial port	Control
6 – Presentation						
5 – Session						
4 – Transport	TCP		UDP			
3 – Network	IP version 4, ICMP, ARP					
2 – Link	IEEE 802.3					
1 – Physical	10Base-T Ethernet (Twisted pair)					

Viola's Ethernet driver, TCP/IP stack, HTTP/Web server and operating system are optimized for embedded devices and implemented with ANSI-C to ensure portability. It is also possible to license the software in source code format.

- Open and layered structure, the augmentation of protocols is straightforward (for example data security protocols)
- Functions in subordination of operating system or without operating system
- Implemented with ANSI-C – portability is straightforward
- Optimized for embedded 8- and 16-bit processors
- Scalable according to processing and memory capacity (among others the number of supported TCP connections)
- Comprehensive documentation and test software to verify the functioning of the device
- Supplied in source code format

3.8 Operation of ESERV-10

Devices using a serial port can be connected to the Internet or to a local network by using the ESERV-10 module. ESERV-10 includes the TCP/IP protocols required by the Internet connection and a Web server when the actual target device necessarily doesn't even recognize that it is connected to Internet. Thus the existing hardware and software don't require modification. With the help of ESERV-10, Internet connectivity can be achieved instantly and easily. ESERV-10 is an independent, updatable module.

The operation of ESERV-10 can be divided into following categories:

- Connection via the Internet to a serial port
- Web server

3.8.1 **Connection via the Internet to a serial port**

ESERV-10 operates as an invisible modem from the serial device's point of view.

The data received from the Internet is controlled by ESERV-10, so that only effective load is directed into the serial port without the Ethernet or TCP/IP parts.

The data from the serial port is encapsulated by ESERV-10 into the TCP/IP protocol and transmitted forward through Ethernet.

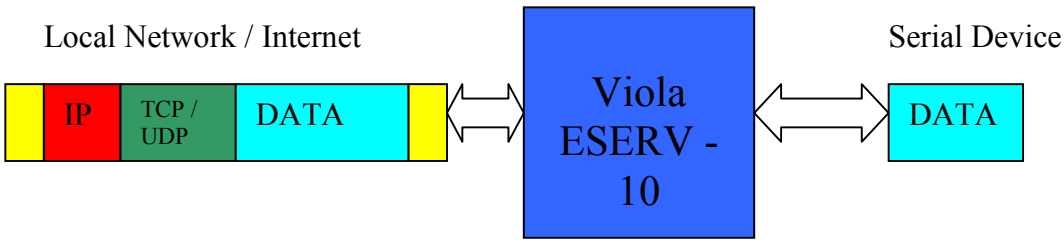


Figure 6. ESERV-10 separates the complexity of the Internet and a serial device

Connection to the serial port can be achieved via the Internet with TCP and UDP through port 7001.

TCP is a so-called **virtual circuit / connection-oriented** protocol that establish a fixed virtual connection between two devices. TCP takes care of the entireness of the data by using checksum and data transmission which use automatic handshakes. Since TCP establishes a fixed connection between devices, the serial port can only be used from one TCP connection at a time. For example when using TELNET program, TCP protocol is used.

UDP is a so-called connectionless protocol that doesn't require specific opening or closing of the connection. UDP also uses checksum to ensure the entireness of the data, but it doesn't automatically take care of retransmission and it doesn't ensure the receiving of data. As a protocol UDP is more insecure but it is also lighter and more general-purpose. Since UDP doesn't require a fixed connection, several Internet devices can use the serial port. ESERV-10 operates so that it transmits all incoming data to UDP port 7001 to the IP address from which a connection has lastly been established. Due to that, ESERV-10's UDP implementation is well suited for question-answer -type data transmission; in case the retransmission feature is already built-in (which is the case in most serial protocols).

As a conclusion one could say that the use of UDP protocol is simpler and even recommended if possible. Some firewalls prevent UDP connections due to security reasons since Windows networks use it to share files. When using such programs as TELNET or if the data transmission should be ensured, TCP may be a better alternative.

In case of both protocols, firewalls may prevent outgoing traffic. In such occasions the firewall must be configured to support the ports and the IP addresses used by the ESERV-10 if one prefers to establish a connection outwards (from the Internet).

3.8.2 **Web server**

ESERV-10 includes a Web server in TCP port 80. HTML pages, figures, documents and Java applets smaller that 130 kB can be downloaded into that.

After this, these files can be accessed by a standard Web browser (Internet Explorer, Netscape Navigator) by writing the ESERV-10's IP address and the filename, like http://195.16.220.69/mypage.html.

If the file isn't available, ESERV-10 will return an error page. If the filename is left away (http://195.16.220.69), ESERV-10 will try to find the default page index.html from its memory and this page is returned if it is found. Otherwise an error page is displayed.

The contents of Web pages can be created by using standard development tools, but before they are downloaded into ESERV-10, separate files must be copied into one file that ESERV-10 recognizes. This can be done with a utility program.

All kinds of files can be downloaded into ESERV-10's memory. There is a Microsoft Access database "mimetypes.mdb" in the utility program's set up directory (for example C:\Program Files\ESERV10\) where all the file types that ESERV-10 recognizes are listed. Database consists of the file type extension (like .txt) and the corresponding MIME-type (like text/plain). More file types can be added into the database.

Web server can be used to save a HTML page, which includes a Java applet. That can be downloaded into the users Web browser and it can be used as a virtual user interface in the following manner:

1. User types ESERV-10's address in the browser's address field
2. ESERV-10 returns the HTML page and the Java applet included
3. The applet runs in the user's browser without using the processing capacity of ESERV-10.
4. The applet makes a TCP or a UDP connection to ESERV-10's port 7001, enabling information transmission over the Internet between the applet and the serial port.

ESERV-10's Web server supports six simultaneous connections. To ensure maximum capacity, the number of objects (pictures, HTML pages, applets) should be limited to six.

4 Installation

The installation and operation of the ESERV-10 Starter Kit consists of the following phases:

- Installation of the utility program in a PC
- Ethernet connection
- Setting up the IP address and the parameters
- Checking the network connection

4.1 Installation of the utility program

The utility program for PC is a tool for configuration and control. With this program the IP addresses of ESERV-10 are controlled, parameters are set up and Web pages are downloaded into ESERV-10's memory. The utility program operates in Windows 95/98, 2000, NT and ME environments.

The utility program is installed by starting a program called **setup.exe** from Viola's ESERV-10 Starter Kit CD.

Installation program installs the ESERV-10 utility program, the uninstall program, the required databases and it also creates an icon in the "Start" menu.

Start the utility program. The following window should be displayed.

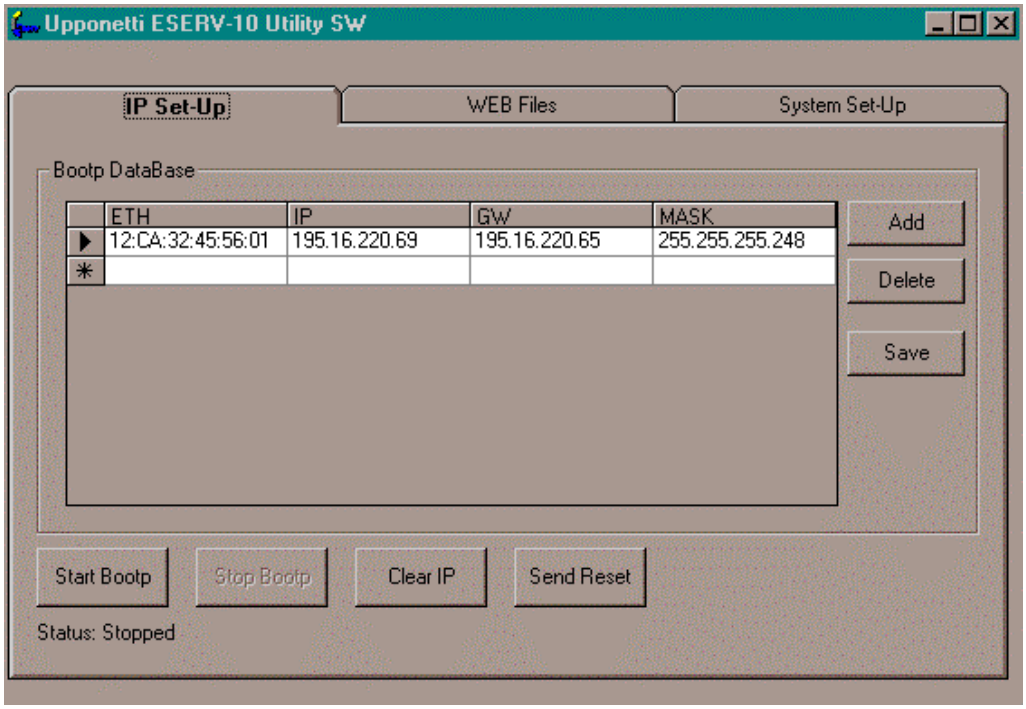


Figure 7. The main window of utility program

4.2 Ethernet connection

Write down the Ethernet/MAC address, which is typed in a sticker under the ESERV-10. The address is of type xx:xx:xx:xx:xx:xx.

Couple the operating voltage and connect the ESERV-10 **10 Mbit/s** to local network with a direct Ethernet cable or to a PC's network interface card with the cross-connected cable included.

The cross-connected cable included with the red connectors is suited only for connecting the ESERV-10 board and a PC's network interface card. If the board is connected to a local network, a direct Ethernet cable should be used.

4.3 Setting up the IP address

When the ESERV-10 is connected to an Ethernet network without an IP address, ESERV-10 will start to send a BOOTP inquiry to the network. A response to this inquiry can be generated with the utility program's BOOTP server. Next the IP address, subnet mask and default router are need to be setup. BOOTP protocol is documented in Internet standards RFC-951 and RFC-1542.

ESERV-10 requires a static IP address and it doesn't support dynamic determination with the help of DHCP. To ensure the successful setup of the IP address, the ESERV-10 and a PC with the utility program must be in same network (same router).

To configure the IP address, the ESERV-10 board's Ethernet/MAC address, the IP address that is given to the Ethernet address, default router and subnet mask are added into the utility program's database. To get the proper addresses, ask your ISP.

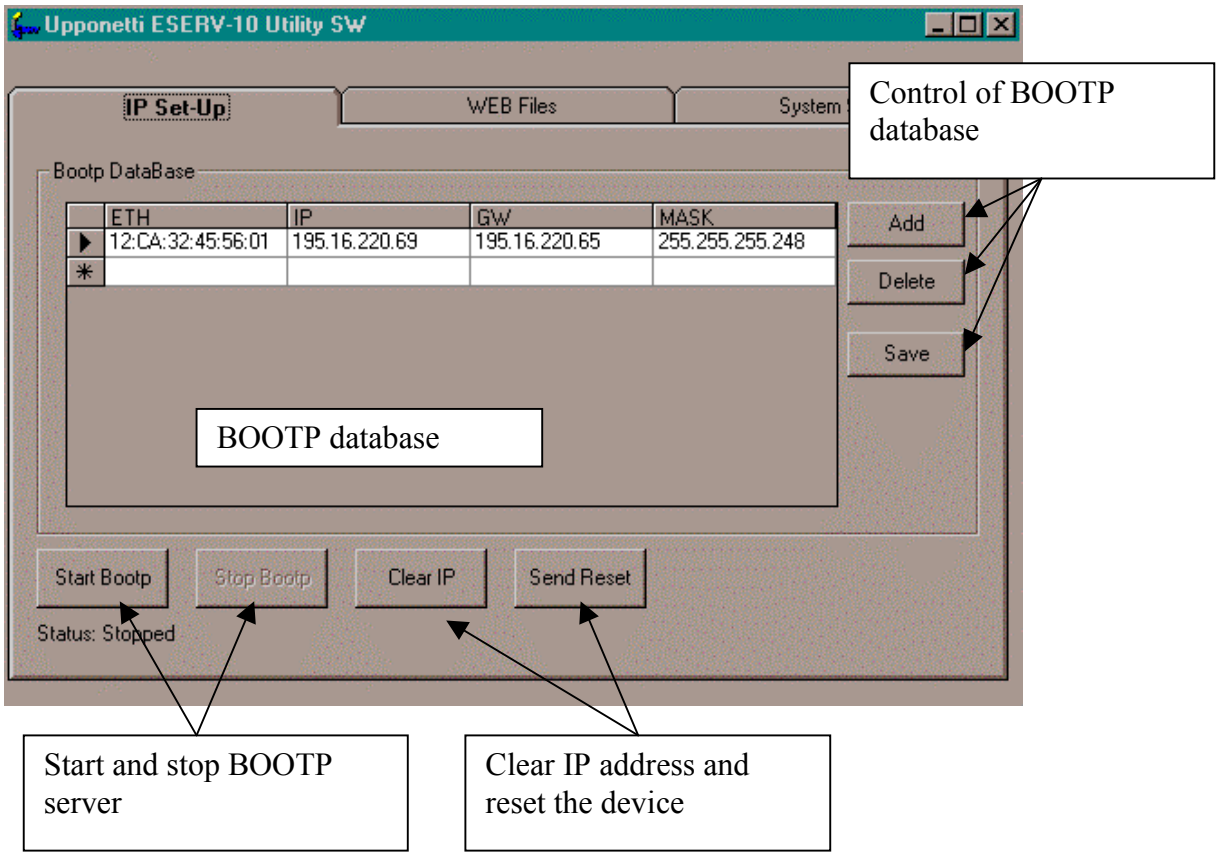


Figure 8. BOOTP server

Type the ESERV-10's Ethernet address, the desired IP address, default router's IP address and subnet mask in the fields of BOOTP database and save the changes by clicking "Save".

After this the BOOTP server can be started by clicking "Start Bootp". If the configuration is successful, a message that tells about sending BOOTP reply will appear in the state line.

To test the IP address configuration write in MS-DOS window ping xxx.xxx.xxx.xxx, where xxx.xxx.xxx.xxx is the ESERV-10's IP address (for example **ping 195.16.220.69**). If configurations has succeeded the reply will be "**Received reply from xxx.xxx.xxx.xxx: bytes=32 time=6 ms TTL=100**".

4.4 Changing the IP address

If there is a need to change the IP address, follow this procedure:

- BOOTP server is interrupted by pressing "Stop Bootp"
- The IP address of the desired ESERV-10 is cleared by pressing "Clear IP" and choosing ESERV-10's current IP address
- BOOTP database is modified so that it will include the new IP address of ESERV-10.
- BOOTP server is started
- A ping inquiry is performed to the new address

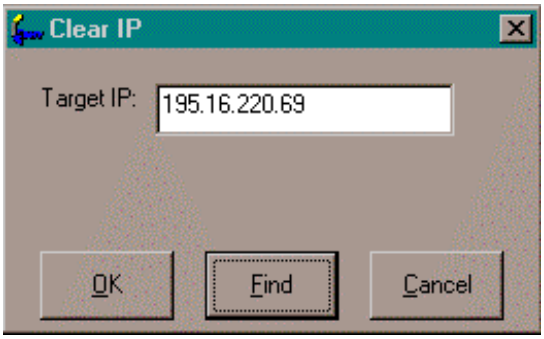


Figure 9. The IP address clearing window

4.5 Testing the operation of the ESERV-10 Web server

The Starter Kit for the ESERV-10 includes a pre-programmed HTML page that can be displayed by writing the ESERV-10's IP address to the browser's address field.

5 Constructing and saving Web pages into ESERV-10's memory

Construction and saving of Web pages include following phases:

- Form files (HTML pages, pictures, applets etc) with normal development tools
- Save all desired files into one directory
- Combine files into one download file (.upn) with the utility software
- Transfer the download file to ESERV-10's memory with the utility software or a TFTP program

5.1 Construction of Web pages

ESERV-10 supports 256 files whose size is at most 130 kB altogether. In Web page development it must be considered that all files are saved in the same directory. ESERV-10 Web server is supposed to transfer simple control pages and applets.

- All files must be located in same directory
- The start page is called **index.html**
- 256 files at most, 130 kB
- Limit the amount of files to six per page
- Favour jpg-pictures due to their small memory demand
- Avoid heavy graphics

5.2 Constructing a download file

Before the pages are transferred into ESERV-10, original files are combined to a download file by using the utility software.

Figure 10 shows the utility software's Web page control window.

The "Build" button starts a process that forms a download file from the files shown in the file window. Finally the program opens a window, which makes it possible to save the download file with an .upn extension.

If all desired files are not shown in the file window, the utility software's installation directory doesn't support the file types concerned. File type extensions (like exe, gif, jpg, doc etc.) and referring MIME-types can be added into the database.

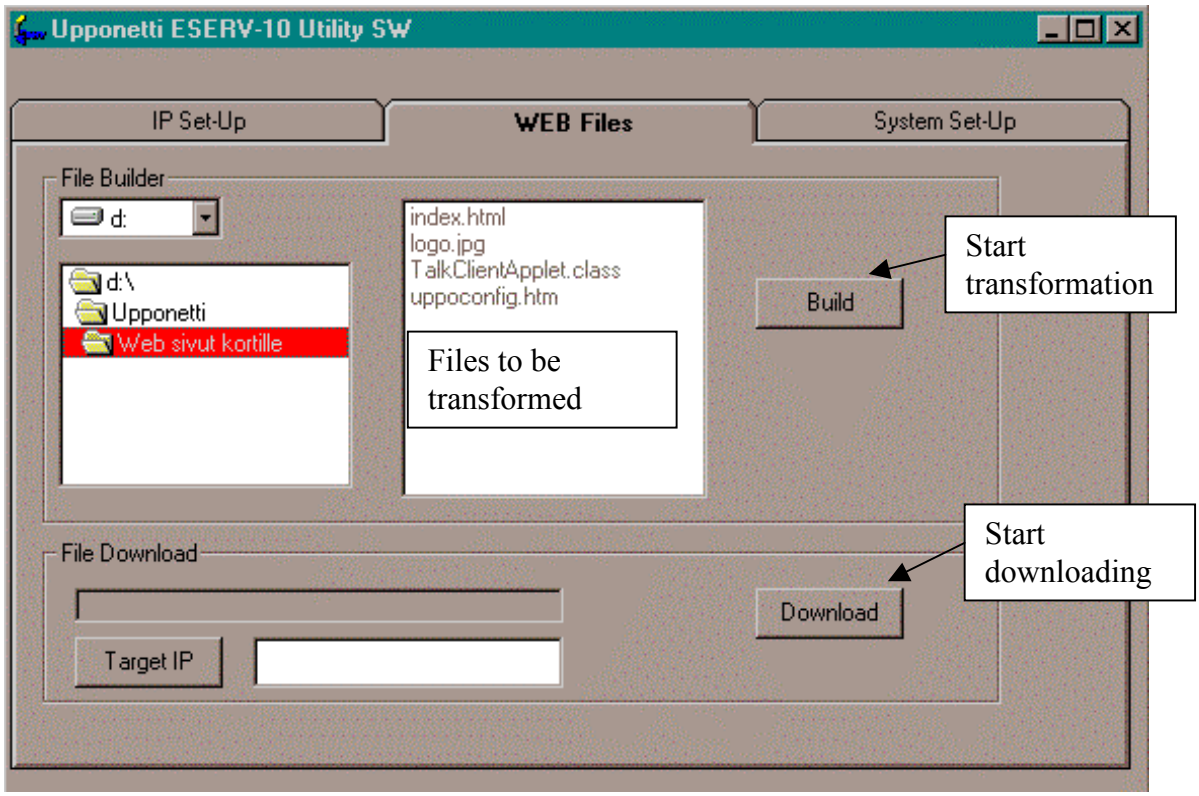


Figure 10. Transformation of Web pages into a download file and transmission of files

5.3 Transmission of download file to ESERV-10

The file is downloaded to the ESERV-10 board by using the utility software's TFTP-feature or some other TFTP program.

If some other TFTP program is used, the target file should be named "ws100.upn". The ESERV-10 utility software takes care of this automatically.

Set ESERV-10's IP address in the download window's address field either by writing or by choosing the address from the database with the "Target IP" button.

File transmission starts with the “Download” button. After that a window opens for choosing a download file. When the file is chosen the transmission starts. A download bar indicates the progress of transmission.

After the transmission is finished, the utility software informs of possible errors or of the process completion.

Type ESERV-10’s address in the Web browser’s address field. New pages should now be downloaded.

5.4 Configuration of parameters

Parameters related to the operation of ESERV-10 can be configured by using the utility software with a device that has an IP address. This can alternatively be performed through a serial port. Few typical values can be chosen with the utility software when parameter setting through a serial port offers more alternatives.

5.4.1 Configuration of parameters by using the utility software

Utility software can be used to configure parameters for serial port operation and network connections for an ESERV-10 device that has an IP address.

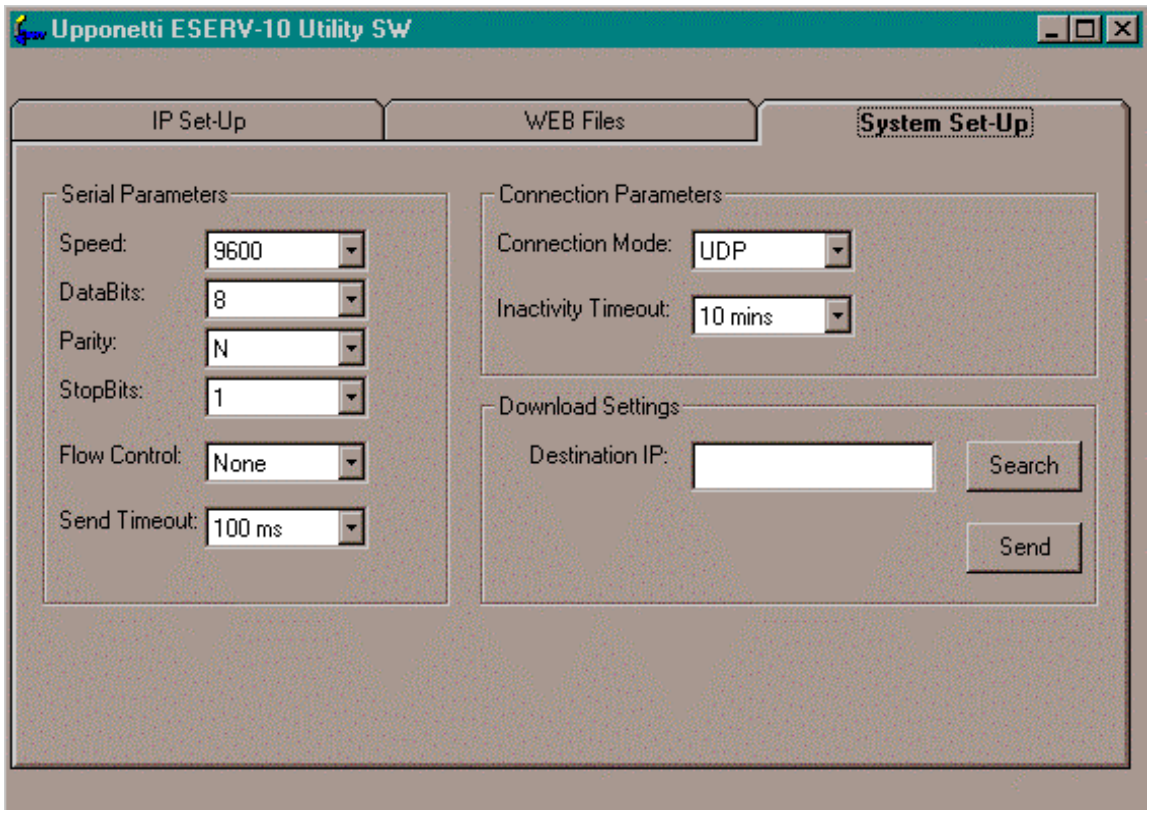


Figure 11. Parameter setup window.

Table 6. Parameter setup using utility software

Parameter	Scale	Purpose
Speed	1200 … 38400 bps	Serial port speed
DatBits	7 or 8	Number of data bits in serial port
Parity	None, Even, Odd	Serial port parity: None = no parity Even = even parity Odd = odd parity
StopBits	1 or 2	Number of stop bits in serial port
Flow Control	None, HW	Serial port flow control None = no flow control HW = CTS/RTS hardware flow control
Send Timeout	0 … 5 seconds	Time that ESERV-10 waits between receiving last byte from serial port and transmitting it to Ethernet
Connection Mode	UDP or TCP	Protocol that enables connection from the Internet to a serial port
Inactivity Timeout	30 seconds … 16 hours	Time that ESERV-10 waits between receiving last packet from Ethernet and closing the connection (TCP).

Flow Control – This feature specifies use of flow control in serial data transmission. Flow control is not necessary from the ESERV-10's point of view if packets received through serial port are less than 512 bytes and data flow isn't continuous. If the amount of data is large, flow control should be used.

Send Timeout – This parameter specifies how long is the time period that ESERV-10 stores the data from serial port before it sends it further to Ethernet. A short time period is advantageous when the data in serial transmission is packet-formed (the whole packet is transmitted as consecutive bytes). A longer time period is preferred if the serial device transmits data gradually.

Connection Mode – This parameter specifies the protocol used to make a connection to ESERV-10's serial port via UDP/TCP port 7001. *Only one of these two protocols can be used.*

Inactivity Timeout – This parameter is a kind of back port when using a serial port with TCP connection. For example if the device (PC) that has a connection with ESERV-10 is unexpectedly closed, **Inactivity Timeout** enables closing the connection if data transmission is terminated. Small Inactivity timeout value is suitable for randomly frequent, short connections, for example when parameters are inquired from a device connected to ESERV-10's serial port. A larger value is suitable when connection is sustained between

ESERV-10's serial port and target device (PC) and data is transmitted continuously. Windows operating system's Inactivity Timeout default value is 1.5 hours.

When a TCP connection exists between the ESERV-10's serial port and the target device, no other connection can be established from any other device. If data transmission is simultaneous and several connections are preferred, UDP must be considered as a reasonable alternative.

ESERV-10 offers the possibility to interrupt the existing TCP connection by sending a "RST" character string to its control port (UDO port 8001). This can be utilized if it is absolutely necessary that one party establishes a TCP connection to a serial port.

5.4.2 Configuration of parameters when using a serial port

Configuration of all ESERV-10 parameters (also IP addresses, default router, subnet mask) can be performed via a serial port.

- Connect ESERV-10's serial port and a PC's serial port with the included null modem cable
- Set the serial data program's (like HyperTerminal) parameters as 9600,8,N,1
- Set the serial data program to transmit <CR><LF> in the end of a line and echo characters locally
- Start ESERV-10 by pressing the board's Reset button
- After one second of ESERV's start press the x-button five times or hold it down when you are starting the operation
- ESERV-10's command request and assistance texts will download into the serial data program's window
- Set the "Caps Lock" button on

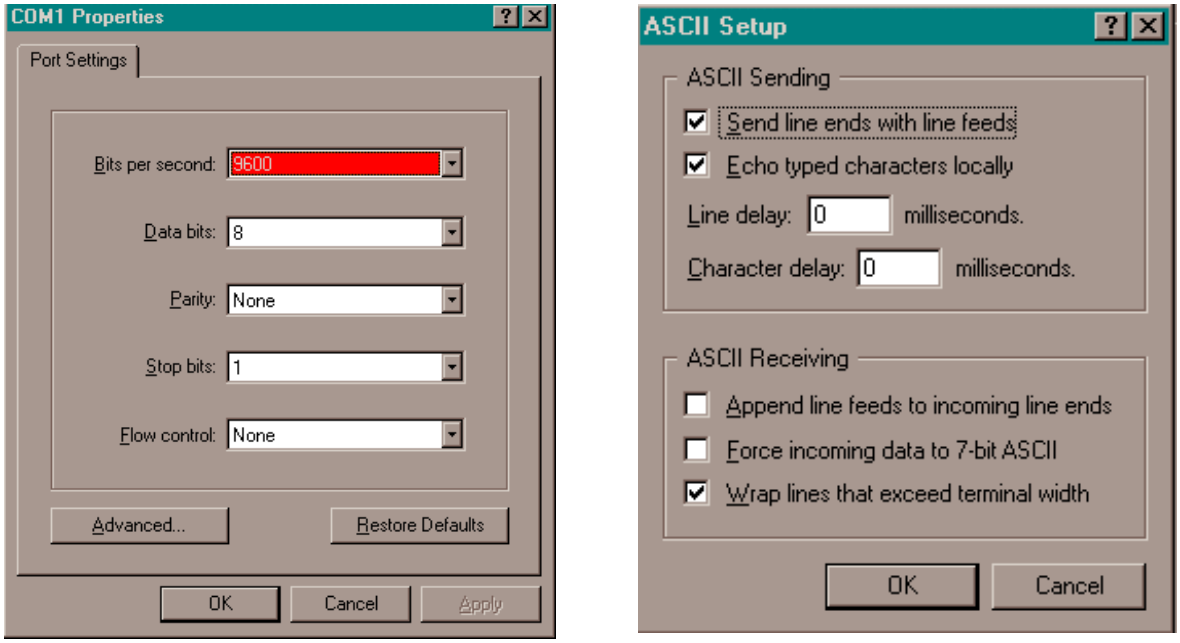


Figure 12. Settings of the serial data program (Hyperterminal)

```
*** Viola ESERV-10 Configuration Mode ***

Change Setup: 0 Rs Speed [1200 ... 38400]
              1 Rs Databits [7,8]
              2 Rs Parity [N,O,E]
              3 Rs Stopbits [1,2]
              4 Rs Flow Control [0=N,1=HW]
              5 Rs Send Timeout[msec]
              6 IP address
              7 Network Mode [0=UDP,1=TCP]
              8 Subnet Mask
              9 Default Gateway
              A Inactivity Timeout [sec]
              B Factory defaults
              C Show Present values

              H HELP (This Screen)
              S Save
              E Save and Exit
              ESC Exit without save

For example: 6=195.16.220.69

OK
ESERV_10:\>
```

Configuration of parameters via a serial port offers more versatile possibilities than configuration by using the utility software. The IP address can also be set via a serial port.

6 Using a serial port connected device over the Internet

The serial port connected device can be accessed through the Internet or a Local Area Network via UDP or TCP port 7001. To do this, a suitable program must be made using C++, Visual Basic, Java etc. It is also possible to use some other software, for example TELNET. On the Starter Kit CD you can find some simple model programs made using Visual Basic and Java.

6.1 Using TELNET

The TELNET client is pre-installed in most of the PC operating systems. To start TELNET in MS Windows operating system, select “Run” from “Start”, type *telnet* and press OK.

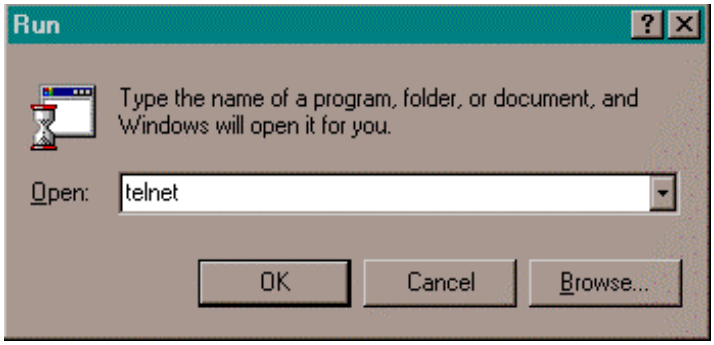


Figure 13. Starting TELNET from Windows

From TELNET select Terminal – Preferences – Local Echo to echo typed characters locally.

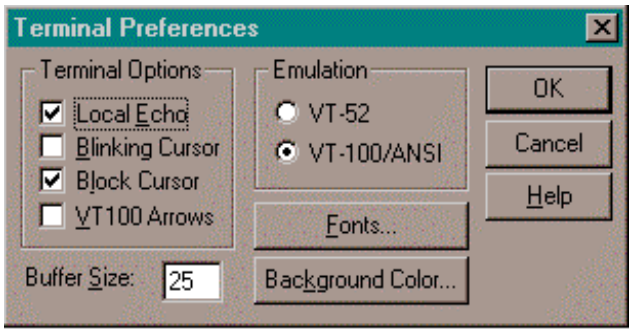


Figure 14. TELNET terminal preferences

When the parameters are configured as described, it is time to open a connection with ESERV-10. From TELNET select Connect – Remote system and type ESERV-10's IP address in the Host Name -field. The port to be used is 7001.

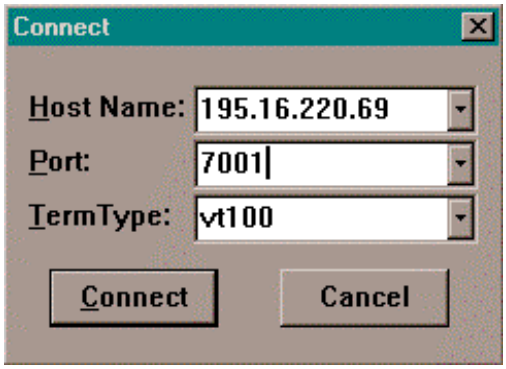


Figure 15. Opening the TELNET connection.

When the connection is successfully established, data can be transferred between the TELNET window and the ESERV-10's serial port.

ESERV-10 MUST be configured for TCP protocol.

6.1.1 TELNET demonstration

A simple serial port using TELNET demonstration can be made by connecting the serial port of ESERV-10 to your PC's serial port by using the null modem cable that is supplied with the Starter Kit. This gives the possibility to observe the data transmission occurring between the Hyper Terminal and TELNET. In this case, the transferred data goes through the Local Area Network and the ESERV-10 into the serial port and vice versa. This is done by utilizing the TCP protocol.

6.2 Internet programming using Visual Basic

Microsoft Visual Basic offers a simple way to construct an Internet protocol utilizing communication using the Winsock – ActiveX object. The interface is also very user friendly, as it is graphic and easy to use. On the Starter Kit CD you can find some simple model programs of TCP and UDP client programs.

6.2.1 Winsock object

The Winsock object includes the support needed for UDP and TCP utilizing operating. It is possible to include the object into the toolbar by selecting Project – Components- Microsoft Winsock Control 6.0 and then by drawing the acquired object into the window (Form).

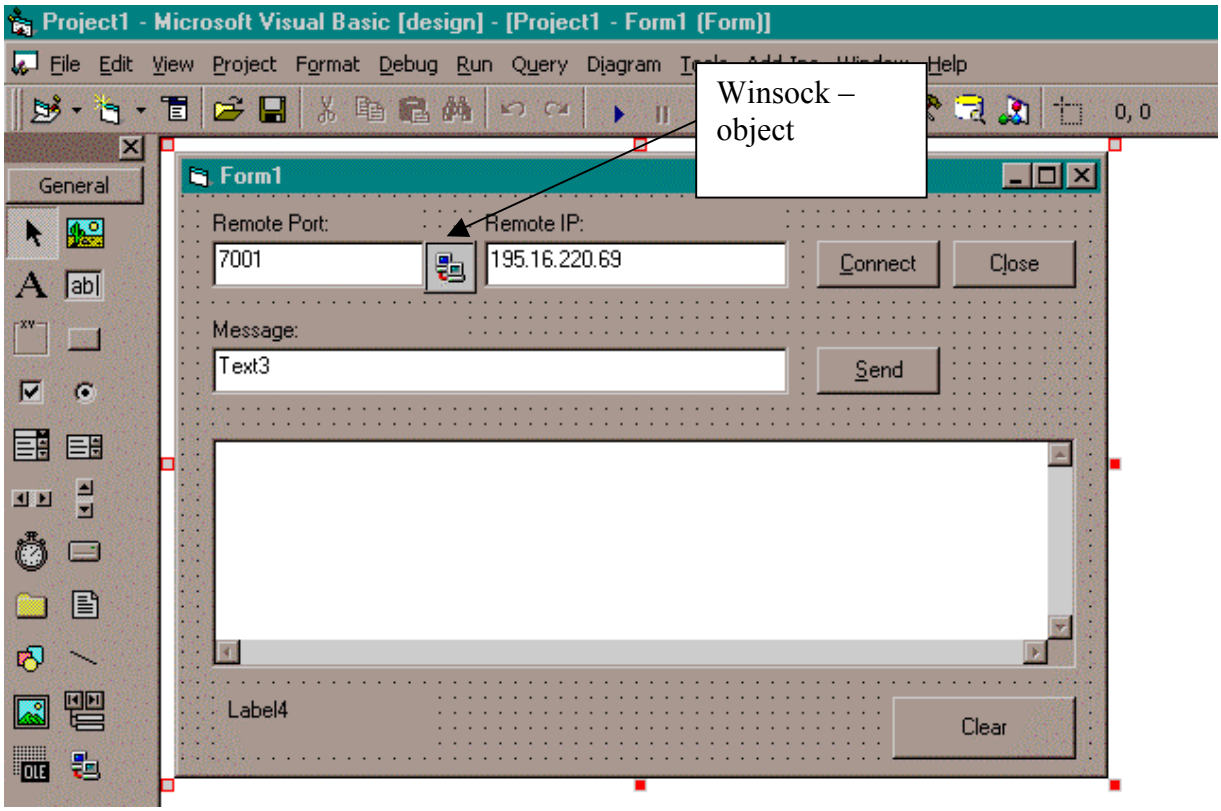


Figure 16. The placed Winsock object

After this the utilizing of Internet protocols is very straightforward. Table 7 shows the most important methods.

Table 7. Most important methods of Winsock control

Method	Parameters	Description
Winsock.Protocol		Sets the protocol utilized by the Winsock control (TCP or UDP)
Winsock.RemotePort	Port	Sets the communications port (e.g. 7001)
Winsock.RemoteIP	IP address	Sets the IP address you wish to be in contact with
Winsock.Connect	IP address, port	Opens a TCP connection to the port of the set IP address
Winsock.SendData	Data	Sends data to the port of the set IP address
Winsock_DataArrival()	BytesTotal	The method that the operating system calls for when data is received. The method includes the value BytesTotal, which indicates the amount of received bytes
Winsock.GetData	Variable, Type	This method is usually used inside the Winsock_DataArrival() function for reading the data received to a given variable.
Winsock.Close	-	Closes the current TCP connection
Winsock_Close()	-	This method is called for by the operating system when the remote peer wants to close the current TCP connection. Usually this request is accepted by the calling method Winsock.Close.

The method and the functions are the same for both TCP and UDP, except for the unavailability of Connect and Close methods for UDP.

7 Using Java applets

It is possible to use a Web server for example to save a HTML page that includes a Java applet. This is loaded into the users Web browser and can be used as a graphic interface for a device in the following manner:

- The user writes the address of the ESERV-10 in the browser's address field
- ESERV-10 returns a HTML page and the Java applet on it
- The applet starts running in the browser and does not use ESERV-10's processing capacity
- The applet establishes a TCP or UDP connection to ESERV-10's port 7001. This makes it possible to transmit data between the applet and a serial port by using the Internet.

An example of a simple Java applet is included on the ESERV-10 CD. The Sun Microsystems' homepage http://java.sun.com also offers information on how to use Java.

8 Using Viola ESERV-10 as client

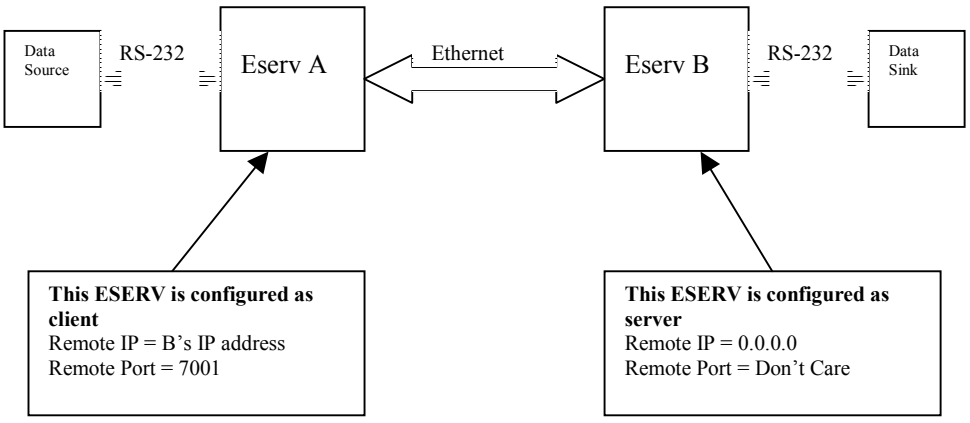
Viola ESERV-10 supports ’client mode’ which means that TCP or UDP communication could be initiated by ESERV-10 to predefined IP address and port. To be able to use ‘client mode’, the following settings must be configured (by using serial configuration mode):

- Right Network Mode you want to use (UDP or TCP)
- Remote IP address (command I)
- Remote port (command P)
- Connection timeout in the case of TCP.

The ‘client mode’ could be disabled by configuring remote IP address to value 0.0.0.0 (default).

Example use of Client Mode

The following example shows a configuration where ESERV-10 labeled A is supposed to initiate connection and send serial data to predefined address of ESERV B (connection is initiated when there appears data to A’s serial port). The configurations for both ESERV-10s are shown in picture.



Viola ESERV-10 S User's Manual

Version 2.0, March 2002
http://www.violasystems.com

© 2002 Viola Systems Ltd.

Viola Systems Ltd.
Lemminkäisenkatu 32
FIN-20520 Turku
Finland

http://www.violasystems.com

E-mail: info@violasystems.com
Tel. +358 201 226 226
Fax. +358 201 226 220

Register number 800.044
Ly 1632065-8